**Exercise 2: E-commerce Platform Search Function**

import java.util.\*;

class Product

{

int productId;

String productName;

String category;

public Product(int productId, String productName, String category)

{

this.productId = productId;

this.productName = productName;

this.category = category;

}

@Override

public String toString()

{

return "ID=" + productId + ", Name=" + productName + ", Category=" + category ;

}

}

public class Main

{

public static Product linearSearch(Product[] products, String targetName)

{

for (Product product : products)

{

if (product.productName.equalsIgnoreCase(targetName))

{

return product;

}

}

return null;

}

public static Product binarySearch(Product[] products, String targetName)

{

Arrays.sort(products, Comparator.comparing(p -> p.productName.toLowerCase()));

int left = 0;

int right = products.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

int comparison = products[mid].productName.compareToIgnoreCase(targetName);

if (comparison == 0)

{

return products[mid];

}

else if (comparison < 0)

{

left = mid + 1;

}

else

{

right = mid - 1;

}

}

return null;

}

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of products: ");

int n = Integer.parseInt(sc.nextLine());

Product[] products = new Product[n];

for (int i = 0; i < n; i++)

{

System.out.println();

System.out.println("Enter details for product " + (i + 1) + ":");

System.out.print("Product ID: ");

int id = Integer.parseInt(sc.nextLine());

System.out.print("Product Name: ");

String name = sc.nextLine();

System.out.print("Category: ");

String category = sc.nextLine();

products[i] = new Product(id, name, category);

}

System.out.println();

System.out.print("Enter product name to search: ");

String searchName = sc.nextLine();

Product linearResult = linearSearch(products, searchName);

if (linearResult != null)

{

System.out.println("Linear Search Result: " + linearResult);

}

else

{

System.out.println("Linear Search: Product not found.");

}

Product binaryResult = binarySearch(products, searchName);

if (binaryResult != null)

{

System.out.println("Binary Search Result: " + binaryResult);

}

else

{

System.out.println("Binary Search: Product not found.");

}

sc.close();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

A computer screen shot of a black screen

AI-generated content may be incorrect.

Time Complexities:

* Linear Search : O(n)
* Binary Search : O(log n)

**Exercise 7: Financial Forecasting**

import java.util.\*;

public class Main

{

public static double predictFutureValue(double presentValue, double growthRate, int years)

{

if (years == 0)

{

return presentValue;

}

return predictFutureValue(presentValue, growthRate, years - 1) \* (1 + growthRate);

}

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter Present Value: ");

double presentValue = sc.nextDouble();

System.out.print("Enter Annual Growth Rate (in %, e.g. 5 for 5%): ");

double growthRatePercent = sc.nextDouble();

double growthRate = growthRatePercent / 100;

System.out.print("Enter Number of Years to Forecast: ");

int years = sc.nextInt();

double futureValue = predictFutureValue(presentValue, growthRate, years);

System.out.printf("Future Value after %d years: %.2f%n", years, futureValue);

sc.close();

}

}

A computer screen shot of a black screen

AI-generated content may be incorrect.

Time Complexity : O(n)